**Experiment Number : 1 Date: 06/01/25**

## **Create a Node.js Module and Import & Use Modules in a Web Server Application**

### **PRE LAB EXERCISE**

**QUESTIONS:**

1. **What are Node.js modules?**

Node.js modules are reusable blocks of code that help organize functionality into separate files and libraries. They can be **built-in**, **third-party**, or **user-defined**. Modules allow developers to keep their code modular, making it easier to manage and reuse.

**Types of Node.js Modules:**

Core Modules , Local Modules , Third-Party Modules

1. **How do you create and export a custom module in Node.js?**

To create and export a custom module in Node.js, follow these steps:

**Define the module** – Create a separate JavaScript file and write the required functions or variables.

**Export the module** – Use module.exports to make the functions or objects available for other files.

**Import the module** – Use require('./moduleName') in another file to access the exported content.

1. **What is the difference between built-in, third-party, and user-defined modules?**

| **Type** | **Definition** | **Example** |
| --- | --- | --- |
| **Built-in Modules** | Pre-installed with Node.js and provide core functionality. | fs, http, path |
| **Third-party Modules** | Installed using npm and developed by the community. | express, mongoose |
| **User-defined Modules** | Custom modules created by developers to organize code. | require('./math') (local files) |

1. **How do you import a module using require()?**

**Syntax:**

const moduleName = require('module-name');

**Built-in Modules** are imported directly by their names (e.g., require('fs')).

**Third-party Modules** are installed via npm and then imported (e.g., require('express')).

**User-defined Modules** are imported using relative paths (e.g., require('./math'))

1. **What is the purpose of the module.exports object?**

In Node.js, the module.exports object is used to define the content that a module exposes to other files. It allows functions, objects, or variables to be shared and reused in different parts of an application.

**Purpose of module.exports:**

* Exports functions, objects, or variables from a module.
* Enables modular programming by separating concerns.
* Allows reusability across multiple files

### **IN LAB EXERCISE**

#### **OBJECTIVE:**

* To understand the creation of Node.js modules.
* To import and use custom and built-in modules in a web server application.

#### **RESOURCES:**

* Node.js installed
* Any text editor (VS Code recommended)
* Basic knowledge of JavaScript

#### **PROGRAM LOGIC & IMPLEMENTATION:**

1. Create a custom module (myModule.js).
2. Export functions from the module.
3. Import and use the module in a Node.js web server (server.js).

### **PROGRAM: Create and Use Node.js Modules**

#### **Step 1: Create a Custom Module (**myModule.js**)**

// myModule.js

exports.greet = function(name) {

return `Hello, ${name}! Welcome to Node.js`;

};```

exports.add = function(a, b) {

return a + b;

};

#### **Step 2: Create a Web Server and Import Module (**server.js**)**

const http = require('http');

const myModule = require('./myModule'); // Import custom module

const server = http.createServer((req, res) => {

res.writeHead(200, {'Content-Type': 'text/plain'});

res.write(myModule.greet('Student')); // Using the greet function

res.write('\nSum: ' + myModule.add(10, 20)); // Using the add function

res.end();

});

server.listen(3000, () => {

console.log('Server running at http://localhost:3000');

});

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

**Output:**

A screenshot of a computer

AI-generated content may be incorrect.

### **POST LAB EXERCISE**

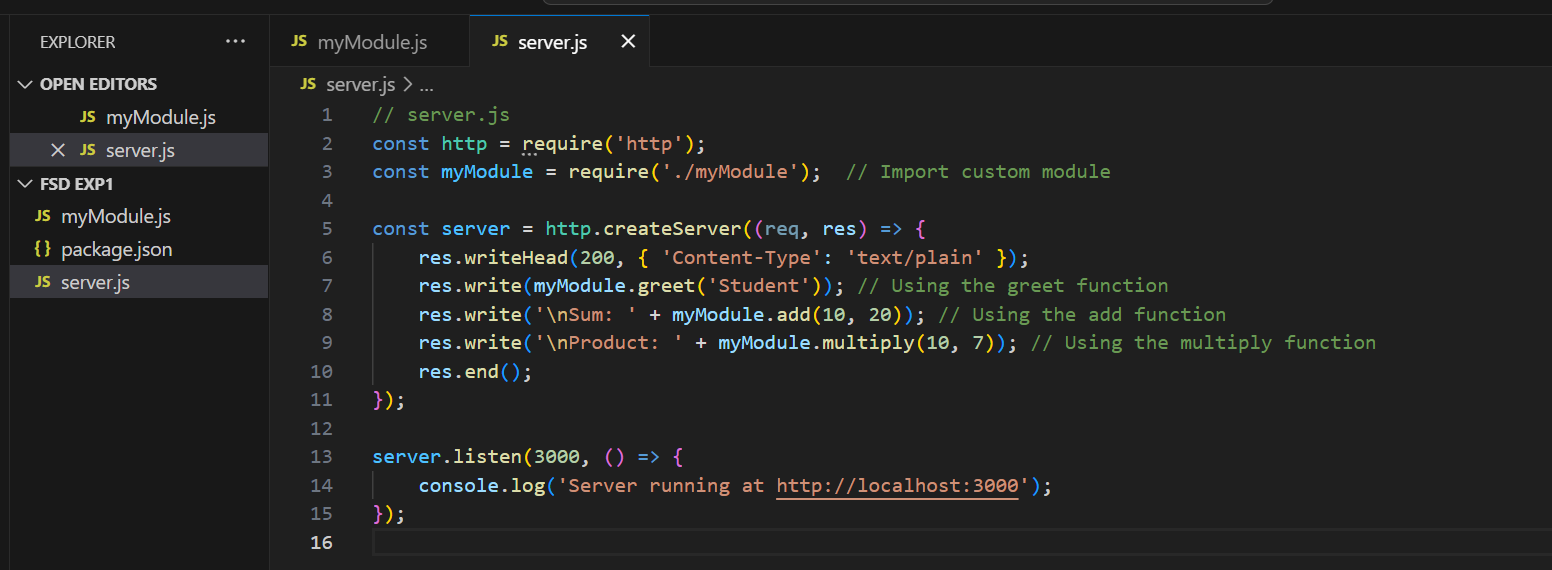
1. **What is the purpose of module.exports in Node.js?**

The purpose of module.exports in Node.js is to define what a module exposes and makes available to other files. It allows functions, objects, or variables to be shared and accessed outside the module, enabling code reusability and modular programming. Without module.exports, the content within a module would remain private to that file.

1. **Modify the myModule.js file to include a function that multiplies two numbers.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

****

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **How does Node.js differ from traditional JavaScript execution in a browser?**

Node.js differs from traditional JavaScript execution in a browser in several ways:

* Environment: Node.js runs on the server-side, while JavaScript in the browser runs on the client-side.
* APIs: Node.js provides server-related APIs like fs for file handling, while browsers provide web-related APIs like document for DOM manipulation.
* Concurrency: Node.js uses an asynchronous event-driven model, allowing non-blocking operations, whereas browsers handle UI interactions and events in the main thread.
* File System Access: Node.js has access to the server’s file system, while browsers do not directly access local files for security reasons.

1. **What happens if you try to import a module that does not exist?**

If you try to import a module that does not exist in Node.js, it will result in an error. Specifically, Node.js will throw a MODULE\_NOT\_FOUND error. This happens because Node.js cannot locate the specified module in the current directory, node\_modules folder, or any other configured paths.

**ASSESSMENT PATTERN.**

|  |  |  |
| --- | --- | --- |
| **Description** | **Max Marks** | **Marks Awarded** |
| Pre Lab Exercise | **5** |  |
| In Lab Exercise | **10** |  |
| Post Lab Exercise | **5** |  |
| Viva | **10** |  |
| **Total** | **30** |  |
| **Faculty Signature** | |  |